REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	5/4/2005	Final Progress Report 2/1/04-1/31/05

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Advanced Formal Methods for Reactive Systems Engineering	DAAD190110019
6. AUTHOR(S) Rance Cleaveland, Eugene Stark, Scott A. Smolka	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Department of Computer Science SUNY Stony Brook Stony Brook, NY 11794-4400	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U. S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSORING / MONITORING AGENCY REPORT NUMBER 41242.1-C1

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT	12 b. DISTRIBUTION CODE
Approved for public release; distribution unlimited.	

13. ABSTRACT (Maximum 200 words)

Significant scientific progress has been made during the final year of the grant. A main accomplishment has been improving the performance of the PIOATool and comparing its performance to the PRISM model checker. We have also designed and implemented the Aristotle runtime verification tool suite and applied it to the Linux kernel, as well as the GMC software model checker for GCC. We also developed and implemented a generic, on-the-fly technique for checking the correctness of real-time systems. .

14. SUBJECT TERMS	15. NUMBER OF PAGES
Probabilistic Input/Output Automata, Runtime Verification, Software Model Checking, Real-Time System Verification	5
	16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**UL**

REPORT DOCUMENTATION PAGE (SF298)
(Continuation Sheet)

1 List of Papers Submitted or Published Under ARO Sponsorship

1. E. Stark, "Formally Specifying CARA in Java," *International Journal on Software Tools for Technology Transfer*, Vol. 5, No. 4, pp. 331-350 (2004).

2. E. Stark and W. Song, "Linear Decision Diagrams." Unpublished technical report, available at `http://bsd7.starkhome.cs.sunysb.edu/~stark/REPORTS/ldd.pdf`

3. R. Grosu and S.A. Smolka. "Safety-Liveness Semantics for UML 2.0." *Proceedings of ACSD 2005: Fifth International Conference on Application of Concurrency to System Design*, IEEE Computer Society Press, Los Alamitos, CA, USA, (June 2005).

4. P. Ye, E. Entcheva, R. Grosu, and S.A. Smolka. "Efficient Modeling of Excitable Cells Using Hybrid Automata." *Proceedings of Computational Methods in Systems Biology*, Lecture Notes in Computer Science, Springer-Verlag (April 2005).

5. R. Grosu and S.A. Smolka. "Monte Carlo Model Checking." *Proceedings of TACAS 2005: Eleventh International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science, Springer-Verlag (April 2005).

6. C.W. Keller, D. Saha, S. Basu, and S.A. Smolka. "FocusCheck: A Tool for Model Checking and Debugging Sequential C Programs." *Proceedings of TACAS 2005: Eleventh International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science, Springer-Verlag (April 2005).

7. P. Yang, Y. Dong, C.R. Ramakrishnan, and S.A. Smolka. "Compiling Mobile Processes for Efficient Model Checking" (Winner of Most Practical Paper Award). *Proceedings of Seventh International Symposium on Practical Aspects of Declarative Languages (PADL 05)*, Lecture Notes in Computer Science, Springer-Verlag (Jan. 2005).

8. S. Basu, D. Saha, and S.A. Smolka. "Localizing Program Errors for Cimple Debugging." *Proceedings of 24th IFIP International Conference on Formal Techniques for Networked and Distributed Systems (FORTE 2004)*, Lecture Notes in Computer Science, Springer-Verlag (Sep. 2004).

9. A. Ray and R. Cleaveland. "An Algebraic Theory of Boundary Crossing Transitions." In G. Luettgen and M. Mendler, editors, *Workshop on the Semantic Foundations of Engineering Design Languages*, Barcelona, Spain, April 2004. Satellite workshop of the European Joint Symposia on Theory and Practice of Software. *Electronic Notes in Theoretical Computer Science*.

10. A. Ray and R. Cleaveland. "Formal Modeling of Middleware-based Distributed Systems." In *Workshop on Formal Foundations of Embedded Software and Component-Based Architecture*,

Barcelona, Spain, April 2004. Satellite workshop of the European Joint Symposia on Theory and Practice of Software. *Electronic Notes in Theoretical Computer Science.*

11. A. Ray, B. Sengupta and R. Cleaveland. "Secure Requirements Elicitation Through Triggered Message Sequence Charts." In R. K. Ghosh and Hrushikesha Mohanty, editors, *Distributed Computing and Internet Technology: First International Conference* (ICDCIT 2004), volume 3347 of Lecture Notes in Computer Science, pages 273-282, Bhubaneswar, India, December 2004. Springer-Verlag.

12. B. Sengupta and R. Cleaveland. "Triggered Message Sequence Charts." To appear in *IEEE Transactions on Software Engineering.*"

13. R. Cleaveland, S. Purushothaman Iyer, and M. Narasimha. "Probabilistic Temporal Logics via the Modal Mu-Calculus." To appear in *Theoretical Computer Science.*

2 Scientific Personnel Supported by the Project

Senior Personnel Scott Smolka, Eugene Stark, Rance Cleaveland

Graduate Students Arnab Ray, Dezhuang Zhang, Zan Sun, Pei Ye and Wenkai Tan

3 Report of Inventions

4 Scientific Progress and Accomplishments

Significant scientific progress has been made during the fourth and final year of grant DAAD190110003 in the following areas: We have continued the development of PIOAL, the process-algebraic specification language for Probabilistic I/O Automata (PIOA) that forms the basis for our tool integration effort, namely, the integration of the PIOATool and the Concurrency Workbench. We have also developed a *Monte Carlo model checking* algorithm, based on the use of random sampling of lassos in Büchi automata; a Hybrid-automaton model of cardiac cells that efficiently captures many essential aspects of the cell's biological behavior; and a safety-liveness semantics for UML 2.0 Sequence Diagrams. We have moreover pursued the development of a mathematical formalism supporting the combined modeling of functional and performance aspects of systems; and the development of a mathematical formalism for software architecture specification.

4.1 Process-Algebraic Language for PIOA

PIOAL is a process-algebraic specification language based on PIOA. We presented PIOAL in a CONCUR 03 paper that describes the new language, its typing rules, and its operational semantics. The paper also presents basic metatheorems relating the typing rules and operational semantics, and establishes congruence properties with respect to probabilistic bisimulation equivalence and PIOA behavior equivalence. Over the past year, we implemented a stand-alone parser and type-checker for PIOAL. In addition, we implemented an algorithm for translating specifications expressed in this

language directly into Linear Decision Diagrams, the matrix-based representation used internally by PIOATool.

We have been looking for axiomatizations of the two equivalences mentioned above for fragments of the language and have succeeded in finding an axiomatization in the case of probabilistic bisimulation. Moreover, a complete axiomatization for PIOA behavior equivalence is nearly finished; this work will be written up for publication before the end of the calendar year.

4.2 Monte Carlo Model Checking

In a TACAS 2005 paper with Radu Grosu, we describe $\texttt{MC}^2$, what we believe to be the first randomized, Monte Carlo algorithm for temporal-logic model checking. Given a specification S of a finite-state system, an LTL formula φ, and parameters ϵ and δ, $\texttt{MC}^2$ takes $M = \ln(\delta)/\ln(1-\epsilon)$ random samples (random walks ending in a cycle, i.e *lassos*) from the Büchi automaton $B = B_S \times B_{\neg\varphi}$ to decide if $L(B) = \emptyset$. Let p_Z be the expectation of an accepting lasso in B. Should a sample reveal an accepting lasso l, $\texttt{MC}^2$ returns false with l as a witness. Otherwise, it returns true and reports that the probability of finding an accepting lasso through further sampling, under the assumption that $p_Z \geq \epsilon$, is less than δ. It does so in time $O(MD)$ and space $O(D)$, where D is B's recurrence diameter, using an optimal number of samples M. Our experimental results demonstrate that $\texttt{MC}^2$ is fast, memory-efficient, and scales extremely well.

We are also in the process of applying Monte Carlo techniques to the model-checking problem for *timed automata*. Our initial results indicate that the performance and scalability advantages of the Monte Carlo approach carry over into the setting of real-time systems.

4.3 Efficient Modeling of Excitable Cells Using Hybrid Automata

This effort is concerned with using Hybrid automata (HA) for efficiently modeling complex biological systems. HA combine discrete transition graphs with continuous dynamics. Our goal is to efficiently capture the behavior of excitable cells previously modeled by systems of nonlinear differential equations. In particular, we derive HA models from the Hodgkin-Huxley model of the giant squid axon, the Luo-Rudy dynamic model of a guinea pig ventricular cell, and a model of a neonatal rat ventricular myocyte. Our much simpler HA models are able to successfully capture the action-potential morphology of the different cells, as well as reproduce typical excitable cell characteristics, such as refractoriness (period of non-responsiveness to external stimulation) and restitution (adaptation to pacing rates). To model electrical wave propagation in a cell network, the single-cell HA models are linked to a classical 2D spatial model. The resulting simulation framework exhibits significantly improved computational efficiency in modeling complex wave patterns, such as the spiral waves underlying pathological conditions in the heart. A description of this work appears in a CMSB (Computational Methods for Systems Biology) paper.

4.4 Safety/Liveness Semantics fo UML 2.0 Sequence Diagrams

We provide an automata-theoretic solution to one of the main open questions about the UML standard, namely *how to assign a formal semantics to a set of sequence diagrams without compromising refinement?* Our solution relies on a rather obvious idea, but to our knowledge has not been used before in this context: that bad and good sequence diagrams in the UML standard should

be regarded as safety and liveness properties, respectively. Proceeding in this manner, we obtain a semantics that essentially complements the set of behaviors associated with the set of sequence diagrams, thereby allowing us to use the standard notion of refinement as language inclusion. We show that refinement in this setting is compositional with respect to sequential composition, alternative composition, parallel composition, and star+ composition. A paper on this work, performed jointly with Radu Grosu, appeared in ACSD 2005.

4.5 Architectural System Modeling

We also continued developing the executable modeling notations developed as part of the original CARA research effort. In one line of work, we gave a thorough algebraic characterization of hierarchical state machines with so-called "boundary-crossing" transitions. Such state machines are very useful in practice, as evidenced by the popularity of the Statecharts notation. However, it was a widely held believe that, like `goto` statements, boundary-crossing transitions inherently "break" system structure and thus cannot be accounted for in a compositional manner. We showed this not to be the case by developing the notion of boundary-crossing transitions as "exception-raising".

We also extended the Architectural Interaction Diagrams (AIDs) software-architecture framework to incorporate notions of security. AIDs permit executable system models to be assembled out of executable components; by including features that regulate information flow, we showed how simulation-based techniques may be used to identify and repair security breaches.

5 Technology Transfer

Cleaveland and Smolka are co-founders, along with Steve Sims, of Reactive Systems, Inc. (RSI), which makes advanced design tools for control-software engineering. RSI's main product is the Reactis tool suite, a companion product to The MathWorks Model-Based design tools. Reactis allows MathWorks users to automatically generate thorough yet compact test suites for Simulink/Stateflow models. It also allows one to visualize the execution of models on generated tests with a highly sophisticated visual simulation environment. The Company is a member of The MathWork's Connections program, and currently has 25 automotive and aerospace customers spread across seven countries. Cleaveland also made over 40 presentations about Reactis to different customers during the year. Part of the technology underpinning Reactis has been influenced by ARO-supported research of Cleaveland and Smolka. To learn more about Reactive Systems, please visit the company web site at `www.reactive-systems.com` or contact Cleaveland or Smolka directly.

In other technology-transfer efforts, Scott Smolka gave a presentation on Monte Carlo model checking at the ARO-sponsored 2004 HCES workshop on High-Confidence Embedded Systems. Cleaveland gave presentations on his and Smolka's experiences in starting Reactis at the 2004 Monterey Workshop in Baden, Austria, and he delivered and invited address on software V&V at the MATLAB EXPO, the premiere model-based software development meeting in Tokyo.